Adaptive Control Design for Underactuated Systems using Sums-of-Squares Optimization

Joseph Moore[1] and Russ Tedrake[2]

***Abstract*— Few methods have been proposed for designing adaptive controllers for underactuated systems with constant, unknown parameters. This is primarily because nonlinear underactuated systems are typically not feedback linearizable; most model-reference adaptive control approaches for nonlinear systems, especially those involving Lyapunov analysis, rely heavily on feedback linearization to guarantee stability. Self-tuning regulators do not have this limitation, but these more flexible methods are often computationally intractable and usually lack a proof of stability.**

Here, we propose an alternative adaptive control design procedure which can handle underactuated systems of moderate dimension ($\leq$ 12). By making use of recent advances in sums-of-squares optimization, we build upon the work done in [31] and [10], to design adaptive controllers with verified robustness to parameter uncertainty, and time-varying adaptive controllers with guaranteed finite-time performance along system trajectories. We demonstrate our algorithm on three simulated underactuated systems - the Acrobot and cart-pole systems with unknown viscous friction terms and the perching glider with unknown aerodynamic coefficients.

I. Introduction

Designing controllers which can reason about parametric uncertainty has long been a major challenge for control systems engineers. Model-based control design techniques which promise higher performance than manually tuned controllers often require very accurate models of the plant dynamics which can be difficult or time-consuming to obtain. For this reason, methods such as robust control and adaptive control have emerged as powerful solutions for handling parameter uncertainty. However, as the system under consideration becomes more complex, it becomes increasingly difficult to design controllers which ensure stability when the parameters are unknown a priori.

It is a particularly challenging task to design adaptive controllers for the class of systems known as *underactuated systems*. Not only are these systems usually nonlinear, but they are also not feedback linearizable, preventing one from applying standard model reference adaptive control methods for nonlinear systems, as has been done for many fully actuated robotic systems in the past[20]. Instead, for underactuated systems, one is often restricted to self-tuning approaches, where parameter estimation and controller design are separated. Usually, this involves using a recursive estimation process to determine the system parameters and a model-based control design algorithm, such as the linear quadratic regulator or pole-placement, to update the controller design based on the most recent parameter estimates[1]. However, while these self-tuning methods are possible to construct, they have a number of significant drawbacks. First of all, the control design algorithms used to generate the control laws for these nonlinear systems (i.e. optimal trajectory design and local linear feedback) are often very computationally intensive, and therefore not amenable to real-time implementation. Secondly, it is well known that self-tuning methods for arbitrary nonlinear systems are difficult, if not impossible to analyze, and mostly lack formal proofs of stability[1].

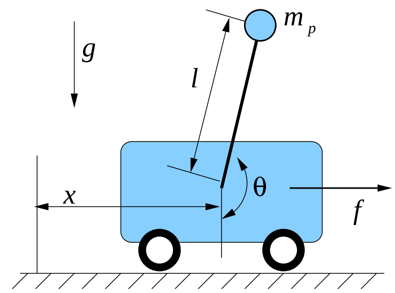

Fig. 1. The cartpole system: a well-known example of an underactuated system. Note that there is no actuation at the pendulum's pivot point.

A third adaptive control design method which has been proposed recently is the *adaptive control Lyapunov function* approach as described in [7]. This method provides a means of finding an adaptive controller for a particular nonlinear system when provided with a special form of Lyapunov function for that system.

In this paper, we use this notion of adaptive control Lyapunov functions as well as recent advances in semi-definite programming to automate adaptive controller design. To this end, we construct a sums-of-squares (SOS) optimization routine and search over a parameterized set of controllers and Lyapunov functions. From this optimization, we are then able find both an adaptive controller for a given nonlinear system as well as its formal certificate of regional stability.

To present the details of this method, we organize our paper as follows. In section II, we summarize some of the work related to the adaptive control of underactuated systems. Then, in section III, we clearly define underactuated systems and what it means for an underactuated system to achieve trajectory following. In section IV, we summarize our approach for time-invariant systems, and present some

This project was supported by ONR MURI Award No: N00014-10-1-0951 and NSF Award No: IIS-0915148

[1]Ph.D. Candidate, Massachussetts Institute of Technology, Cambridge, MA 01239, USA (joemoore@mit.edu)

[2]Associate Professor of Eelectrical Engineering and Computer Science, Massachussetts Institute of Technology, Cambridge, MA 01239, USA (russt@mit.edu)

results on the Acrobot system balancing about the upright. Then, in section V, we extend our results to time-varying systems, and apply our methods to the cart-pole system and the perching glider system. In section VI we describe a method of formulating the SOS optimization problem so that our method can handle dynamics which are non-affine in the unknown parameters.

II. Related Work

In recent years, underactuated systems have become the focus of many research initiatives, since underactuation is a prominent feature in many practical systems. Walking robots, where the contact between the foot and the ground is an unactuated joint, and fixed-wing aerial vehicles in complicated flow regimes, are two clear examples of where reasoning about underactuation is essential[28]. A wide variety of control methods have been applied to these underactuated systems, from optimal trajectory design[2], [3], to energy methods [25], to real-time planning[8], [6], to sliding mode control[15], [14], and partial feedback linearization[24], [23], [22]. More recently, sums-of-squares verification has been applied to these underactuated systems to provide certificates of stability or finite-time performance for local controllers designed based on a linearization around the nominal solution[29], [30]. In the time-varying case, these certificates take the form of time-varying Lyapunov functions, or "funnels", which certify a set of initial conditions that is guaranteed to arrive at the goal in finite-time. In [11], these funnels were extended to the case where the goal region is variable, and in [26], the funnels were extended to the case where stochastic uncertainty is present. In [29], [13], these funnels were combined to cover a larger region of state space by forming a LQR-Tree.

There are relatively fewer methods for handling underactuated systems in adaptive control design. In [18], the authors address adaptive control of underactuated systems by applying partial feedback linearization to the collocated joint and carrying out region of attraction (ROA) analysis on the non-collocated joint. In [19], the authors apply backstepping to find a suitable Lyapunov function with which they can design an adaptive controller for their autonomous underwater vehicle. Similarly, in [4], the authors develop an adaptive controller for an underactuated quadrotor UAV capable of compensating for uncertain mass.

Here we follow the approach described in [7], which introduces the adaptive control Lyapunov function and an approach to designing these functions using back-stepping. Using this Lyapunov function, the authors are then able to derive a nonlinear adaptive controller in a manner similar to [21]. In this paper we replace the back-stepping approach by automating the search for the controller and Lyapunov functions using SOS optimization.

III. Metrics for Robust Performance in Underactuated Systems

Our focus in this paper is on underactuated mechanical systems, defined as[28]:

Definition 1: Consider the control affine system $\ddot{x} = f_1(x,\dot{x}) + f_2(x,\dot{x})u$. We define this system to be *underactuated* in state $(x,\dot{x})$ if $\text{rank}[f_2(x,\dot{x})] < \dim[x]$. We say that the system is underactuated if it is underactuated in any state.

Two well-known examples of underactuated mechanical systems include the cart-pole and Acrobot systems[25], which are simple two degree-of-freedom robotic manipulators each with only a single actuator. The rank condition which defines the underactuated systems implies that the standard approach to feedback linearization, which requires $f_2^{-1}(x,\dot{x})$, cannot be applied. Furthermore, the standard approaches to model-reference adaptive control for mechanical systems (c.f., [20]) which use error-driven feedback to overcome model errors and ensure stability before the parameters are identified are no longer valid.

Without feedback linearization, designing adaptive controllers with provable stability properties becomes more complex. Given a desired trajectory, $x_d(t)$, defined for $t \in [t_0, \infty]$, and unknown model parameters, a natural goal is to design an adaptive control law for which we can prove that the closed-loop system converges asymptotically to this desired trajectory for all possible parameters in some family. However, with no guaranteed ability to directly cancel parameteric uncertainty using feedback, for some systems it may be difficult or impossible to achieve asymptotic convergence to a desired trajectory. Furthermore, we would argue that for most real-world tasks this level of performance is not necessary.

Alternatively, we can define the goal of control as converging to a bounded invariant region in the vicinity of the desired trajectory, and attempt to maximize the set of initial conditions from which the controller provably achieves this goal. Following [30], we call these bounded invariant regions "funnels" and define them formally as follows:

Definition 2: Given the closed-loop dynamics $\dot{x} = f(t,x)$ with $x \in \mathbb{R}^n$, a set $\mathscr{F} \subset [t_0,t_f] \times \mathbb{R}^n$ is a funnel if for each (τ, x_τ) in $\mathscr{F}$, the solution to $\dot{x} = f(t,x)$ with $x(\tau) = x_\tau$ exists on $[\tau, t_f]$ and for each $t \in [\tau, t_f]$ we have $(t, x(t)) \in \mathscr{F}$.

Note that these funnels can be defined over either a finite horizon or an infinite horizon, $t_f = \infty$. For finite-duration desired trajectories, we often find it useful to demand even less of the controller, requiring only that it arrives in a goal region, $G \subset \mathbb{R}^n$, at the final time, t_f. In the case of unknown model parameters with bounded uncertainty, we will require that under the closed-loop dynamics given by the adaptive controller, this funnel into the goal is achieved for all parameters of interest.

We examine both formulations in the following two sections.

IV. Control Design for Asymptotic Convergence

We first consider the problem of designing an adaptive controller which is guaranteed to drive the system asymptotically to a desired trajectory. For clarity, we restrict the presentation in this section to regulating a time-invariant system to the origin; the extension to time-varying systems and desired trajectories is straight-forward and will be presented in the next sections.

A. Adaptive control Lyapunov functions

Following [7], we consider the system

$$\dot{x} = f(x) + F(x)\theta + g(x)u, \quad (1)$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $\theta \in \mathbb{R}^p$, and search for an adaptive controller with the form

$$u = \alpha(x, \hat{\theta}) \quad (2)$$
$$\dot{\hat{\theta}} = \tau(x, \hat{\theta}), \quad (3)$$

where $\hat{\theta}$ is an estimate of the parameter θ.

Now consider a candidate Lyapunov function for this system

$$V(x, \theta, \hat{\theta}) = V_a(x) + \frac{1}{2}\tilde{\theta}^T \Gamma \tilde{\theta}, \quad (4)$$

where V_a is a smooth, positive-definite, scalar function, $\tilde{\theta} = \theta - \hat{\theta}$ and $\Gamma = \Gamma' \succ 0$.

Taking the time derivative of this candidate Lyapunov function, we have

$$\begin{aligned}\dot{V} &= \frac{\partial V_a(x)}{\partial x}\dot{x} + \tilde{\theta}^T \Gamma \dot{\tilde{\theta}} \\ &= \frac{\partial V_a(x)}{\partial x}(f(x) + F(x)\theta + g(x)\alpha(x, \hat{\theta})) + \tilde{\theta}^T \Gamma \tau(x, \hat{\theta}).\end{aligned}$$

If we then let

$$\dot{\hat{\theta}} = \tau(x, \hat{\theta}) = -\Gamma^{-1}\left(\frac{\partial V_a(x)}{\partial x}F(x)\right)^T, \quad (5)$$

we have

$$\dot{V} = \frac{\partial V_a(x)}{\partial x}(f(x) + F(x)\hat{\theta} + g(x)\alpha(x, \hat{\theta})).$$

Thus, if we can find a $V_a(x)$, Γ, and $\alpha(x, \hat{\theta})$ such that $\frac{\partial V_a(x)}{\partial x}(f(x) + F(x)\hat{\theta} + g(x)\alpha(x, \hat{\theta})) < 0$, $\forall x, \theta, \hat{\theta}$, then we have a global adaptive controller which guarantees that $x \to 0$ as $t \to \infty$ by Barbalat's Lemma. Furthermore, if these conditions are met over an invariant region, $(x, \theta, \hat{\theta}) \in B \subset \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p$, then we obtain asymptotic convergence to the origin for all initial conditions in B. Note that any θ in B is trivially invariant (θ is assumed stationary by definition), but verifying the Lyapunov conditions for θ in a subset of $\mathbb{R}^p$ provides a means to guarantee the controller's performance over a set of possible (unknown) parameters for the system. For this reason, we will consider adaptive control Lyapunov functions of the form:

$$V = V_a(x) + \frac{1}{2}\tilde{\theta}^T \Gamma \tilde{\theta} + \frac{1}{2}\theta^T \Psi \theta$$

with $\Psi = \Psi^T \succ 0$.

B. Sums-of-squares optimization

In recent years, sums-of-squares (SOS) optimization routines[16] have provided a means for automating the search for Lyapunov functions. For dynamical systems with polynomial vector fields, SOS makes it possible to verify the Lyapunov conditions ($V > 0, \dot{V} < 0$) for a candidate Lyapunov function as a convex optimization[16]. It is also possible to search for a Lyapunov function which satisfies the conditions, to search for a Lyapunov function which proves stability in a bounded invariant region, and/or to simultaneously optimize a controller in order to maximize a the verified invariant region by solving a series of convex optimizations[9], [17], [27], [33]. Typically it is difficult to find globally valid controllers for interesting control problems, so in this section we illustrate the application of SOS optimization to the problem of verifying the stability of our affine control Lyapunov function approach over an invariant region.

We restrict our search to positive-definite Lyapunov functions, $V(x, \theta, \hat{\theta})$, and define the region of interest, B, as a sub-level set of this positive function:

$$B = \{(x, \theta, \hat{\theta}) | V(x, \theta, \hat{\theta}) < \rho\},$$

for some positive scalar ρ. Note that if the Lyapunov conditions are satisfied, then this B is an invariant region of the closed-loop system. It is therefore sufficient for regional stability to verify $\forall x, \theta, \hat{\theta} \in B$ that:

$$V(x, \theta, \hat{\theta}) > 0 \quad (6)$$
$$\dot{V} = \frac{\partial V_a(x)}{\partial x}(f(x) + F(x)\hat{\theta} + g(x)\alpha(x, \hat{\theta})) < 0 \quad (7)$$

If $f(x)$, $F(x)$, and $g(x)$ are polynomials, then one observes that all of these constraints are polynomial. In practice, many systems of interest have polynomial (or rational polynomial) vector fields, including most robotic manipulators after a change of variables, or can be efficiently approximated using polynomials.

In SOS optimization, we attempt to prove positivity over B by applying the S-procedure and replacing the test for positivity with the condition that the polynomial is a sum-of-squares. Using Σ to denote the sums-of-squares polynomials over x, θ, and $\hat{\theta}$, we write

$$V + s_0(V - \rho) \in \Sigma \quad (8)$$
$$-\dot{V} + s_1(V - \rho) \in \Sigma \quad (9)$$
$$s_0, s_1 \in \Sigma \quad (10)$$

where s_0, and s_1 are additional polynomials which serve as multipliers for the S-procedure. We seek to maximize the size of B in order to provide a controller that works for the largest possible range of parameters and initial conditions; here we approximate this volume by enforcing a scale on V and maximizing ρ. In order to optimize these polynomial constraints over the decision variables, V_a, α,ρ, s_1, Γ and Ψ, we must carry out three steps of bilinear alternations.

If we let $V_a = x^T S_a x$, for some $S_a = S_a^T \succ 0$, and we let $\Gamma = \Gamma^T \succ 0$ and $\Psi = \Psi^T \succ 0$, then the constraint in equation 8 is satisfied trivially. Note that higher-degree Lyapunov functions could provide more richness, but at the cost of requiring extra SOS constraints to confirm the positivity of V_a. Using $\dot{V} = \frac{\partial V_a(x)}{\partial x}(f(x) + F(x)\hat{\theta} g(x)\alpha(x, \hat{\theta}))$, we can perform the optimization in the following steps:

STEP 1:

$$\begin{aligned} \underset{s_1,\gamma}{\text{maximize}} \quad & \gamma \\ \text{subject to} \quad & \gamma > 0, \\ & -\dot{V} + s_1(V-\rho) - \gamma \in \Sigma. \\ & s_1 \in \Sigma \end{aligned} \tag{11}$$

STEP 2:

$$\begin{aligned} \underset{V,\Gamma,\Psi,\rho}{\text{maximize}} \quad & \rho \\ \text{subject to} \quad & \rho > 0, \\ & -\dot{V} + s_1(V-\rho) \in \Sigma. \end{aligned} \tag{12}$$

STEP 3:

$$\begin{aligned} \underset{\alpha,\rho}{\text{maximize}} \quad & \rho \\ \text{subject to} \quad & \rho > 0, \\ & -\dot{V} + s_1(V-\rho) \in \Sigma \end{aligned} \tag{13}$$

These steps then repeat until convergence is observed in ρ.

C. Results

Once developed, this controller design algorithm was tested on the Acrobot (see figure 2) for the balancing task.

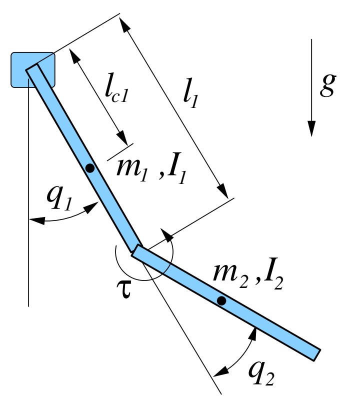

Fig. 2. The Acrobot system: The only actuation is at the elbow joint.

The equations of motion of the Acrobot have trigonometric terms in them; since these appear simply it is possible to either perform an exact change of coordinates to a polynomial vectorfield, or to approximate the dynamics using Taylor expansion. We use a Taylor expansion to third order in the results reported here. For this problem, we parameterized the controller as $\alpha(x,\hat{\theta}) = K_{LQR}x + \hat{\theta}K_{\theta}x$, where K_{LQR} comes from the solution for the linear quadratic regulator for the nominal system when $q = q_0$. θ was specifically chosen to be the (scalar) unknown damping coefficient on *noncollocated* (unactuated) shoulder joint.

To initialize our iterations, we choose $S_a = S$, where S comes from the LQR Riccati equation. We also choose ρ to be very small and $\det(\Gamma)$ and $\det(\Psi)$ to be very large. This is a reasonable initialization procedure because $V = x^T Sx$ is a valid Lyapunov function with some robustness in a small neighborhood around the nominal point. The multipliers were chosen to be up to 4th order.

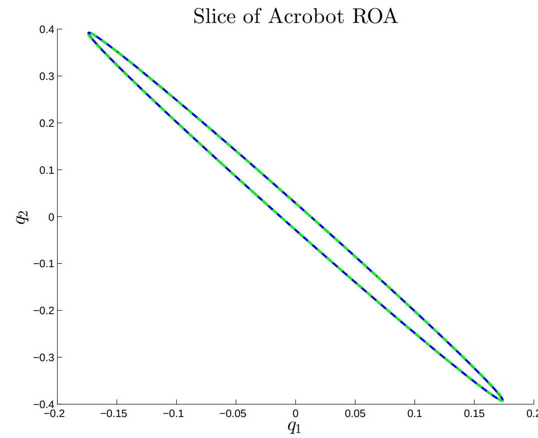

Fig. 3. Slice of four dimensional Acrobot region of attraction in the q_1-q_2. Notice that there is little difference between the size of the ROA for the LQR controller (green) and the adaptive controller (blue)

The results for searching for the controller can be observed in Figure 3, plotted as a slice of the 4 dimensional ellipsoidal level-set $V = \rho$. While non-negligible gains K_θ were found by the optimization procedure, these gains had little affect on the size of the region of attraction. We believe that this is due to the favorable robustness properties of LQR for a time-invariant system.

This being said, there are cases where unknown constant parameters can cause LQR to fail to achieve asymptotic stability. In these cases, we claim that our methods can design an adaptive controller which enables a region of attraction to exist. Consider the case of a constant, unknown offset appearing in the measurements provided to a LQR Controller. We can write these closed loop dynamics as

$$\dot{x} = f(x) + g(x)(K(x+\theta)). \tag{14}$$

Since K is a constant matrix, we can rewrite this as

$$\dot{x} = f(x) + g(x)\alpha(x,\hat{\theta}) + g(x)K\theta \tag{15}$$

which is identical to equation 1 when $F(x) = g(x)K$. To test our approach, we again implemented this case using the Acrobot. We applied a single measurement offset to the Acrobot's shoulder joint and designed and verified our adaptive controller using SOS. The results can be seen in figures 4, 5, and 6. The ROA for the this adaptive controller is comparable to the ROAs shown in figure 3 above while the ROA for the system using the LQR Controller does not exist.

V. Control Design to a Trajectory

In this section, we revisit the control experiment for the cart-pole where the goal of control is specified by a nominal trajectory which swings the robot from the downward configuration to the upright configuration. While asymptotic convergence and convergence to an invariant funnel around

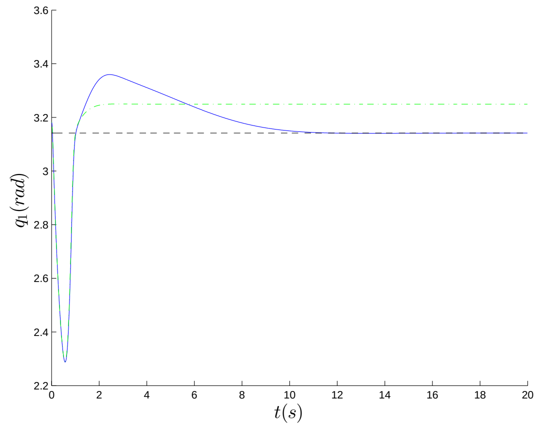

Fig. 4. A plot of q_1 with time. Notice how the LQR Controller (green) fails to converge to π, while the adaptive controller (blue) does.

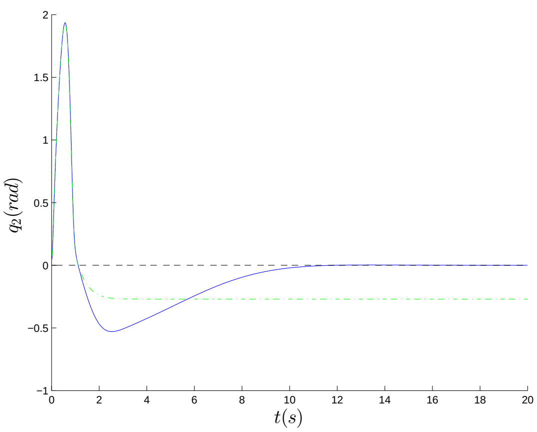

Fig. 5. A plot of q_2 with time. Notice how the LQR Controller (green) fails to converge to zero, while the adaptive controller (blue) does.

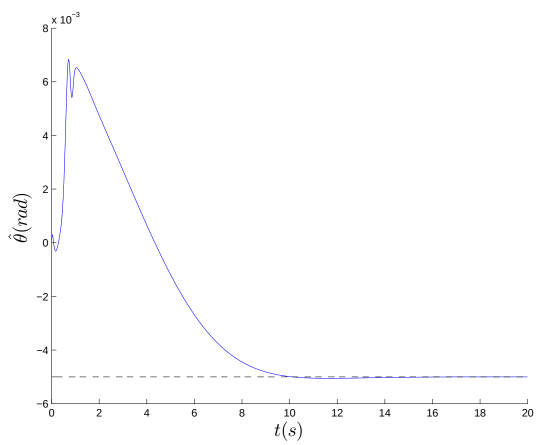

Fig. 6. A plot of $\hat{\theta}$ with time. Even though it is not guaranteed to converge, the estimated parameter (blue) does converge to the true parameter (black).

the trajectory are both possible, due to space limitation we limit our presentation to the formulation that is unique to finite-time trajectories - designing/verifying a controller which provably drives the system to a goal region, G, at a specified final time.

A. Trajectory Design for the Nominal System

Before we carry out the adaptive control design process, we first must find a suitable trajectory for the nominal system by which to parameterize the state-space of interest. Despite the relative complexity of our models with the nonlinear and highly underactuated dynamics, standard tools for trajectory optimization work well for designing a nominal, locally optimal trajectory. To find an optimal trajectory for our systems, we chose to use a direct collocation method [32] implemented using SNOPT [5], which allows us to efficiently encode hard constraints on the states and inputs.

B. Trajectory Stabilization for the Nominal System

To stabilize our nominal trajectory, we use a time-varying linear quadratic regulator (LQR). To apply this control methodology, we linearize our nonlinear system about the nominal trajectory to obtain a time-varying linear system which can be represented as:

$$\dot{\bar{x}}(t) = A(t)\bar{x}(t) + B(t)\bar{u}(t), \tag{16}$$

where $\bar{x}(t) = x(t) - x_0(t)$ and $\bar{u}(t) = u(t) - u_0(t)$. Let us define the following cost function for the maneuver:

$$J = \bar{x}(t_f)^T Q_f \bar{x}(t_f) + \int_0^T \bar{x}(t)^T Q\bar{x}(t) + \bar{u}(t)^T R\bar{u}(t). \tag{17}$$

LQR design yields the control law

$$\bar{u}(t) = -R^{-1}B^T S(t)\bar{x}(t), \tag{18}$$

and the cost-to-go

$$J = \bar{x}^T S(t)\bar{x}, \tag{19}$$

where

$$\dot{S}(t) = Q - S(t)BR^{-1}B^T S(t) + S(t)A + A^T S(t), \tag{20}$$

$$S(t_f) = Q_f. \tag{21}$$

C. Sums of Squares Optimization

For time-varying systems, we are required to slightly modify our sums of squares formulation. Our SOS inequality becomes

$$\begin{aligned} &-\frac{\partial V_a(t,x)}{\partial x}(f(x) + F(x)\hat{\theta}g(x)\alpha(t,x,\hat{\theta})) + \ldots \\ &-\frac{\partial V_a(t,x)}{\partial t} - \tilde{\theta}^T\dot{\Gamma}(t)\tilde{\theta} - \theta^T\dot{\Psi}(t)\theta + \dot{\rho}(t) + \ldots \\ &s_1(t,x,\theta,\hat{\theta})(V_a(t,x) + \frac{1}{2}\tilde{\theta}^T\Gamma(t)\tilde{\theta} + \frac{1}{2}\theta^T\Psi(t)\theta - \rho(t)) > 0 \end{aligned}$$

To handle the dependence in time, we sample in time along the trajectory as is done in [30] and carry out the same three part bilinear optimization scheme as described previously.

D. Controller Parametrization

From simulation, we noticed that the offset between the desired trajectory for the true system and the nominal trajectory seemed to impact the controller performance to a much greater degree than the difference between the ideal time-varying LQR gains and those obtained from the nominal system. Because our control requirement is not exact trajectory following but staying in the funnel while moving between two regions of state space, we found it advantageous to allow for a $\hat{\theta}$-dependent shift of the nominal trajectory,

$$x_d = x_0 + w(\hat{\theta}), \quad u_d = u_0 + v(\hat{\theta}),$$

where $w(\hat{\theta})$ and $v(\hat{\theta})$ are a function that we can potentially design along with the controller.

Consider, as before, the system

$$\dot{x} = f(x) + g(x)u.$$

If $\tilde{x} = x - x_d$, $\tilde{u} = u - u_d$, and $\tilde{u} = K\tilde{x}$, then we can write

$$\dot{\tilde{x}} = f(x) + g(x)(K\tilde{x} + u_d) - \dot{x}_d$$
$$x = f(x) + g(x)(K(x - x_0 - w(\hat{\theta})) + u_0 + v(\hat{\theta})).$$

Now, if $\bar{x} = x - x_0$, then

$$\dot{\bar{x}} = f(x) + g(x)(K(\bar{x} - w(\hat{\theta})) + u_0 + v(\hat{\theta})) - \dot{x}_0$$
$$\dot{\bar{x}} = f(x) + g(x)(K\bar{x} + u_0 - Kw(\hat{\theta}) + v(\hat{\theta})) - \dot{x}_0.$$

For simplicity, here we fix $w(\hat{\theta}) = \gamma\hat{\theta}$ and $v(\hat{\theta}) = \psi\hat{\theta}$ and let $K_\theta = K\gamma - \psi$, then we have

$$\dot{\bar{x}} = f(x) + g(x)(K\bar{x} + u_0 - K_\theta\hat{\theta}) - \dot{x}_0.$$

and thus achieve the controller parametrization $u = K\bar{x} + u_0 - K_\theta\hat{\theta}$, where $K = K_{LQR}$. This allows us to adjust the nominal trajectory using $\hat{\theta}$ and thus provide the controller with a more reasonable x_d to follow.

E. Results

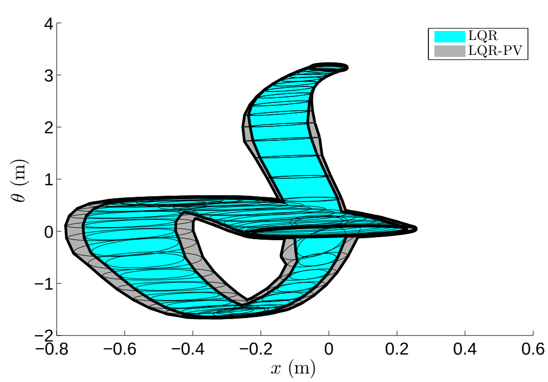

Fig. 7. Slice of four dimensional cartpole funnel in the x-θ plane. The gray region is the controller which makes use of gain-scheduling.

To test our algorithm, we applied it to the cart-pole system (see figure 1) swing up task. Once again, we chose the viscous friction on the non-collocated joint as our uncertain parameter. The resulting funnels for this nominal system

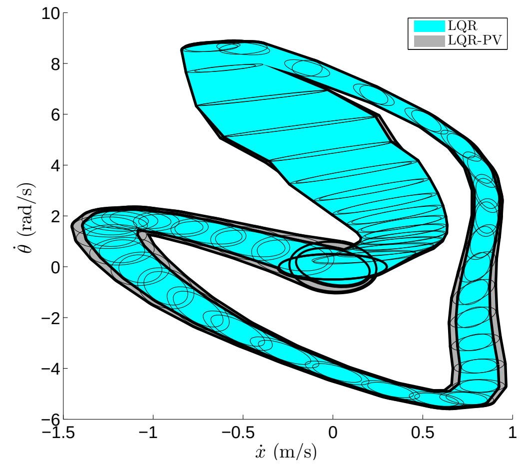

Fig. 8. Slice of four dimensional cartpole funnel in the $\dot{x}$-$\dot{\theta}$ plane. The gray region is the controller which makes use of gain-scheduling.

and gain scheduled system can be see in figures 7 and 8. Note that the gain scheduled funnel is slightly larger than the funnel for the nominal system. In figure 9, we demonstrate the improved performance of the gain-scheduled and adaptive controllers for the cart-pole swing up task. Both the gain-scheduled (known θ) and adaptive (unknown θ) controllers reach the goal region for a greater range of parameter variation than the time-varying LQR controller.

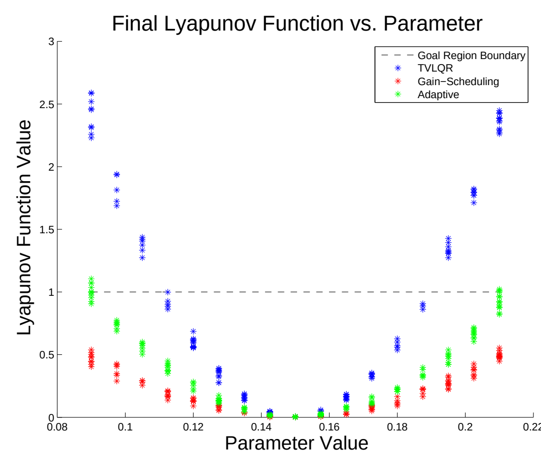

Fig. 9. Plot showing the cartpole system simulated from a range of initial trajectories and parameter values. Both the gain-scheduling controller and the adaptive controller provide an advantage of the TV-LQR design.

We also tested our adaptive control design algorithm on a more advanced, perching glider system (see figure 10), by making the aerodynamic lift coefficient unknown. Figures 11 and 12 demonstrate the clear improvement in performance.

VI. Non-affine in the Parameters

In many instances, an unknown constant parameter will enter in to a system's dynamics in a non-affine manner. For example, consider the case of a constant speed wind gust which enters into an aircraft's aerodynamic forces quadratically. The previously described approach, as it is restricted to

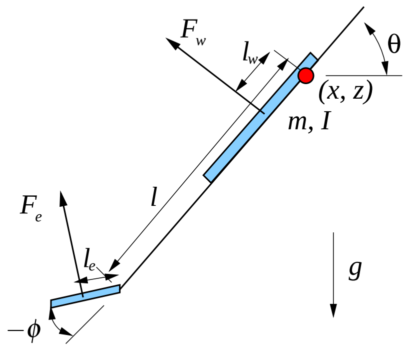

Fig. 10. The perching glider system: the elevator is the only actuator and flat plate lift and drag coefficients are used to compute the aerodynamic forces.

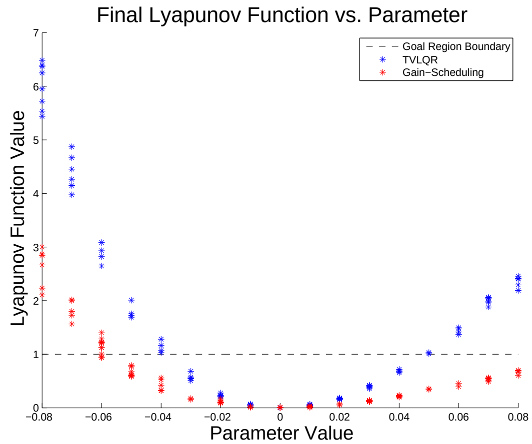

Fig. 11. Plot showing the perching glider system simulated from a range of initial trajectories and parameter values. Both the gain-scheduling controller (red) and the adaptive controller provide an advantage over the TV-LQR design (blue).

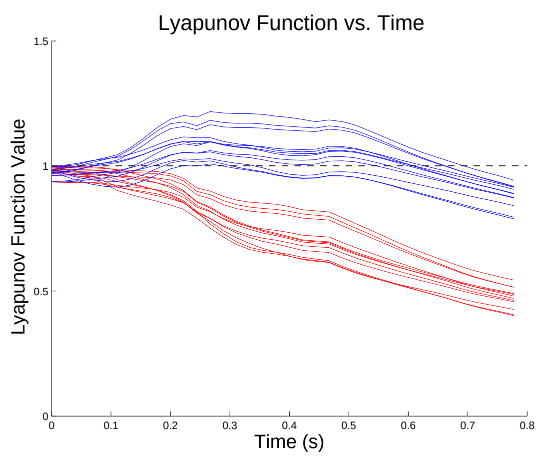

Fig. 12. Plot showing the perching glider system simulated from a range of initial trajectories and parameter values. This plot shows trajectories evaluated along the funnel over time when the $\theta = 0.034$. It is interesting to note that the LQR controller (blue) leaves the funnel around $t \approx 0.2s$ while the adaptive controller (red) does not.

dynamics which are affine in the unknown parameters, can not be applied to this more complex uncertainty structure.

For this reason, we propose an alternative law for $\dot{\hat{\theta}}$ which allows the unknown parameter to be estimated even when it enters the dynamics nonlinearly. We achieve this by requiring that $\alpha(x, \hat{\theta})$ is affine in $\hat{\theta}$.

Consider the system

$$\dot{x} = f(x, \theta) + g(x)\alpha(x, \hat{\theta})$$

and the Lyapunov function

$$V = V_a + \frac{1}{2}\tilde{\theta}^T \Gamma \tilde{\theta} + \frac{1}{2}\theta^T \Psi \theta$$

where $\tilde{\theta} = \hat{\theta} - \theta$. Taking the derivative, we then have

$$\dot{V} = x^T S_a(f(x, \theta) + g(x)\alpha(x, \hat{\theta})) + \dot{\tilde{\theta}}^T \Gamma \tilde{\theta}.$$

If we choose $\alpha(x, \hat{\theta}) = K_{LQR}x + z(x)\hat{\theta}$, we can then write

$$\begin{aligned}\dot{V} &= x^T S_a f(x, \theta) + x^T S_a g(x)(K_{LQR}x + z(x)\hat{\theta}) + \dot{\tilde{\theta}}^T \Gamma \tilde{\theta} \\ &= x^T S_a f(x, \theta) + x^T S_a g(x) K_{LQR}x + x^T S_a g(x) z(x)\hat{\theta} + \dot{\tilde{\theta}}^T \Gamma \tilde{\theta}\end{aligned}$$

Choosing $\dot{\hat{\theta}}^T = -x^T S_a g(x) z(x) \Gamma^{-1}$ and remembering that $\dot{\tilde{\theta}}^T = \dot{\hat{\theta}}^T$ we have

$$\dot{V} = x^T S_a f(x, \theta) + x^T S_a g(x) K_{LQR}x + x^T S_a g(x) z(x)\theta \leq 0.$$

An inspection of $\dot{V}$ reveals that SOS verifcation will produce an adaptation law which can be derived even when θ enters the dynamics nonlinearly. Although we do not explore this formulation any further here, we note the result's importance in demonstrating that the SOS methods described here can be applied to a wide class of adaptive control problems.

VII. Conclusion

In this paper, we have presented a method for designing adaptive controllers for underactuated systems. By using the notion of a "funnel" we were able to use sums of squares optimization techniques to design adaptive controllers which are able to transition an underactuated system between an initial and final condition set for a larger range of uncertain parameter values than standard time-varying LQR. To confirm this performance improvement, the control design technique was applied to both the classical cart-pole system as well as the perching glider system and tested successfully in simulation.

In the future, one of our first tasks will be applying this algorithm to systems with multiple uncertain parameters. Although in theory the algorithm extends naturally to such systems, the impact of additional parameters on numerical conditioning is unclear. Another potential area of future work involves exploring how bounded disturbances impact the control law. The adaptive control algorithm presented here assumes that the the only thing unknown about the system is its constant parameters. There is no question that bounded uncertainty will exist in any real system. Thus, it will be important to understand how this phenomena impacts performance. To help mitigate the effects of uncertainty, we propose incorporating robustness bounds into the SOS verification procedure, as was done in [12] for non-adaptive systems. Similarly, there is a great deal of room for improving the trajectory design process. Nominal trajectories could be designed so that they maximize system excitation or so that they maximize robustness to uncertainty in a particular parameter. Last of all, we hope to explore combining these funnels into an adaptive LQR-Tree.

While there are still many aspects of the algorithm which need to be explored further, the algorithm itself provides a sufficiently general approach for designing adaptive controllers for the important case of underactuated systems. In the near future, we plan to test the adaptive control laws which emerge from this design method on actual hardware, since only then will the utility of the method be able to be truly ascertained.

REFERENCES

[1] Astrom, K.J., Wittenmark, and B. *Adaptive control*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1994.

[2] Marcel Bergerman and Yangsheng Xu. Optimal control sequence for underactuated manipulators. *IEEE Int. Conference on Robotics and Automation*, 1996.

[3] Gary Boone. Minimum-time control of the acrobot. volume 4, pages 3281–3287. IEEE International Conference on Robotics and Automation (ICRA), 1997.

[4] Mu Huang; Bin Xian; Chen Diao; Kaiyan Yang; Yu Feng. Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping. *American Control Conference*, June 2010.

[5] Philip E. Gill, Walter Murray, and Michael A. Saunders. *User's Guide for SNOPT Version 7: Software for Large -Scale Nonlinear Programming*, February 12 2006.

[6] S. Jung and J.T. Wen. Nonlinear model predictive control for the swing-up of a rotary inverted pendulum. *ASME J. on Dynamics, Measurements, and Control*, 126:666–673, September 2004.

[7] Miroslav Krstic and Petar V. Kokotovic. Control lyapunov functions for adaptive nonlinear stabilization. *System and Control Letters*, 26, December 1995.

[8] Lizarralde, F.; Wen, and J.T.; Liu Hsu. A new model predictive control strategy for affine nonlinear control systems. In *Proceedings of the 1999 American Control Conference*, volume 6, pages 4263–4267, San Diego, California, June 1999. IEEE.

[9] Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. *arXiv preprint arXiv:1210.0888*, 2012.

[10] Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[11] Anirudha Majumdar, Mark Tobenkin, and Russ Tedrake. Algebraic verification for parameterized motion planning libraries. In *Proceedings of the 2012 American Control Conference (ACC)*, 2012.

[12] Joseph Moore, Rick Cory, and Russ Tedrake. Robust post-stall perching with a simple fixed-wing glider using lqr-trees. *Under review*, 2014.

[13] Joseph Moore and Russ Tedrake. Control synthesis and verification for a perching uav using lqr-trees. *In Proceedings of the IEEE Conference on Decision and Control*, December 2012.

[14] M. Nikkhah, H. Ashrafiuon, and F. Fahimi. Sliding mode control of underactuated biped robots. In *Proceedings of the ASME IMECE*. ASME, Nov 5-11 2005.

[15] Mehdi Nikkhah, Hashem Ashrafiuon, and Kenneth R. Muske. Optimal sliding mode control for underactuated systems. In *Proceedings of the 2006 American Control Conference*, June 14-16 2006.

[16] Pablo A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, May 18 2000.

[17] Prajna, S., Parrilo, P.A., Rantzer, and A. Nonlinear control synthesis by convex optimization. *IEEE Transactions on Automatic Control*, 49(2):310 – 314, feb. 2004.

[18] Ian Charles Rust. Control of a nonlinear underactuated system with adaptation, numerical stability verification, and the use of the lqr trees algorithm. *S.B. Thesis*, May 2010.

[19] Santhakumar, M., Asokan, and T. Non-linear adaptive control system for an underactuated autonomous underwater vehicle using dynamic state feedback. *International Journal of Recent Trends in Engineering*, 2(5), November 2009.

[20] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, October 1990.

[21] E. Sontag. Feedback stabilization of nonlinear systems, 1989.

[22] M. W. Spong. Underactuated mechanical systems. In B. Siciliano and K. P. Valavanis, editors, *Control Problems in Robotics and Automation*, Lecture notes in control and information sciences 230. Springer-Verlag, 1997.

[23] Mark Spong. Partial feedback linearization of underactuated mechanical systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 314–321, September 1994.

[24] Mark W. Spong. Swing up control of the acrobot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2356–2361, 1994.

[25] Mark W. Spong. Energy based control of a class of underactuated mechanical systems. In *Proceedings of the 1996 IFAC World Congress*, 1996.

[26] Jacob Steinhardt and Russ Tedrake. Finite-time regional verification of stochastic nonlinear systems. *International Journal of Robotics Research*, 31(7):901–923, June 2012.

[27] W. Tan and A. Packard. Searching for control lyapunov functions using sums of squares programming. *Allerton conference on communication, control and computing*, pages 210–219, 2004.

[28] Russ Tedrake. *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines: Course Notes for MIT 6.832*. Working draft edition, 2012.

[29] Russ Tedrake, Ian R. Manchester, Mark M. Tobenkin, and John W. Roberts. LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052, July 2010.

[30] Mark M. Tobenkin, Ian R. Manchester, and Russ Tedrake. Invariant funnels around trajectories using sum-of-squares programming. *Proceedings of the 18th IFAC World Congress, extended version available online: arXiv:1010.3013 [math.DS]*, 2011.

[31] U. Topcu, A.K. Packard, P. Seiler, and G.J. Balas. Robust region-of-attraction estimation. *IEEE Transactions on Automatic Control*, 55(1):137 –142, Jan 2010.

[32] Oskar von Stryk. Users guide for dircol: A direct collocation method for the numerical solution of optimal control problems, November 1999.

[33] Fen Wu and Stephen Prajna. A new solution approach to polynomial lpv system analysis and synthesis. *Proceeding of the 2004 American Control Conference*, June 2004.